📘 React Function Props – Passing and Using in Child Components

---

🔧 Use-Case:

> Passing a function (event handler) from a parent component (App.jsx) to a child component (First.jsx), and executing that function on a button click inside the child.

🧩 App.jsx

```
import { useState } from 'react';
import First from './components/First';
import './App.css';

function App() {
  const [count, setCount] = useState(0);

  function clickHandler() {
    setCount(count + 1);
  }

  function clickRemover() {
    if (count > 0) {
      setCount(count - 1);
    }
  }

  return (
```

```
    <>
      <First clickHandler={clickHandler} clickRemover={clickRemover}>
        <h1>{count}</h1>
      </First>
    </>
  );
}

export default App;
```

🧩 First.jsx

```
import React from 'react';

const First = (props) => {
  return (
    <div>
      <p>{props.children}</p>
      <button onClick={props.clickHandler}>Click me!</button>
      <button onClick={props.clickRemover}>Click to Remove</button>
    </div>
  );
};

export default First;
```

📌 Concept Used: Function as Props

Feature	Description
`clickHandler`	Function to increment count — passed from `App` to `First`.
`clickRemover`	Function to decrement count — passed from `App` to `First`.
`props.children`	Displays the current value of `count`.

📦 Why is this Important?

✅ Function Props Enable:

- Parent component **controls the state**.
- Child component **triggers the change**.
- Great for **component reuse** and **separation of concerns**.

⚙️ Working Flow:

```
App.jsx
  ↳ holds state and functions (count, setCount, clickHandler, clickRemover)

First.jsx
  ↳ receives props:
     - clickHandler
     - clickRemover
     - children (h1 showing count)

  ↳ Executes those functions on button click
```

💡 Best Practices

Tip	Reason
✅ Keep state in the parent	For better control and debugging
✅ Pass only required functions	Keeps components lightweight
❌ Avoid logic duplication	Don't repeat the same logic in both parent and child

---

🚀 Extended Idea (Optional for Learning):

You can pass any type of function — like API calls, toggle actions, modals — not just counter logic.

📝 Summary

Concept	Usage
`useState`	Manage `count` state in parent
Function Props	Send `clickHandler` and `clickRemover` to child
`props.children`	Send JSX (count `<h1>`) to child for rendering
Buttons in child	Trigger parent logic on click
